

# .NET MAUI



## Activitats avaluables 03

MAUI

Piero Zavala Chira | Desenvolupament d'interfícies | 18/10/2024

## Contenido

Introducción .....	2
App.xaml.cs .....	2
MainPage.xaml .....	3
MainPage.xaml.cs .....	4
Page2.xaml .....	5
Page2.xaml.cs .....	6
Pag3.xaml .....	7
Page4.xaml .....	8
Page4.xaml.cs .....	9
Page5.xaml.cs .....	10
Page5.xaml .....	10
Repositorio GitHub .....	11
Webgrafía .....	11

## Introducción

Diseñamos una aplicación con NET MAUI usando NavigationPage.

## App.xaml.cs

Primero tenemos que crear una instancia de NavigationPage en el archivo App.xaml.cs. De esta forma, establecemos MainPage como página principal de la pila de páginas.

A screenshot of a code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in C# and defines a partial class App that inherits from Application. The class has a constructor that calls InitializeComponent() and sets the MainPage property to a new NavigationPage containing a new MainPage instance.

```
namespace CompraNavigationPage
{
    public partial class App : Application
    {
        public App()
        {
            InitializeComponent();

            MainPage = new NavigationPage(new MainPage());
        }
    }
}
```

## MainPage.xaml

En MainPage.xml creamos el diseño

Atributos de botón:

```
<Button
  x:Name="btEmpezarCompra"
  Clicked="btEmpezarCompra_Clicked"
  Text="Comenzar compra"
  SemanticProperties.Hint="Counts the number of times you click"
  HorizontalOptions="Fill" />
```



## MainPage.xaml.cs

Como vimos en MainPage.xaml, inicializamos el atributo Clicker con el nombre «btEmpezarCompra\_Clicked». Se nos creará un método en MainPage.xaml.cs en donde estará la lógica de la página Main. Pondremos un Navigation.PushAsync para ir a la Page2 cada vez que se haga clic en el botón de «Empezar a comprar».

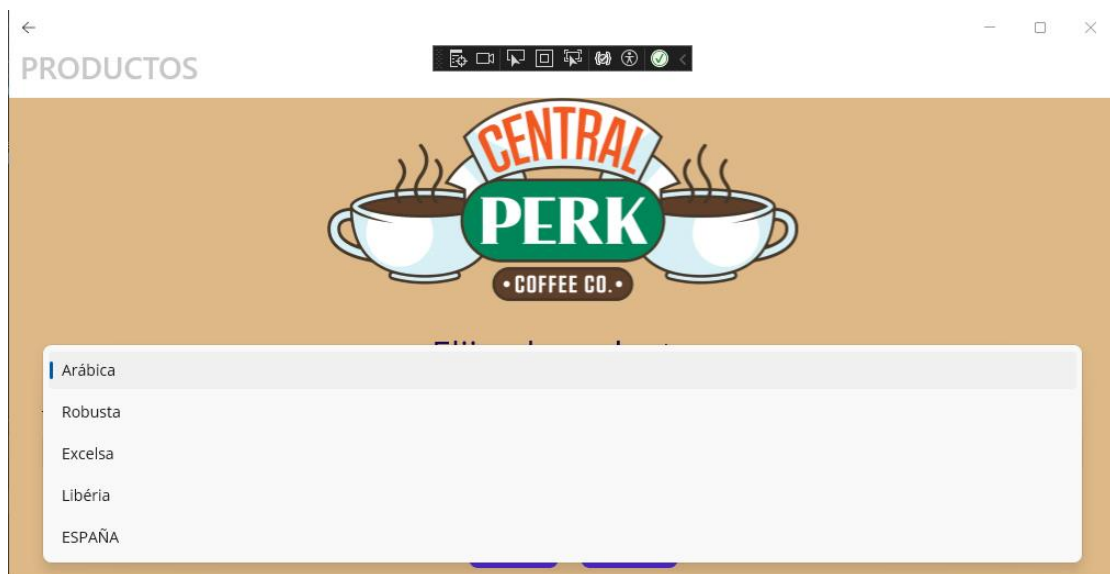
```
namespace CompranavigationPage
{
    public partial class MainPage : ContentPage
    {
        int count = 0;

        public MainPage()
        {
            InitializeComponent();
        }

        private async void btEmpezarCompra_Clicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new Pages.Page2());
        }
    }
}
```

## Page2.xaml

Como podemos ver tenemos una barra de selección (Piker).



A continuación, se muestra el fragmento de código del Picker. Cuando inicializamos el evento SelectedIndexChanged, se crea un atributo en Page2.xaml.cs para poder gestionar la información.

```
<Picker x:Name="pickerProductos"
        Title="Tipo de cafe"
        SelectedIndexChanged="pickerProductos_SelectedIndexChanged">
    <Picker.ItemsSource>
    <x:Array Type="{x:Type x:String}">
        <x:String>Arábica</x:String>
        <x:String>Robusta</x:String>
        <x:String>Excelsa</x:String>
        <x:String>Libéria</x:String>
        <x:String>ESPAÑA</x:String>
    </x:Array>
    </Picker.ItemsSource>
</Picker> the number of times you click"
```

## Page2.xaml.cs

Aquí podemos ver cómo gestionar el método Picker guardándolo en una cadena de texto para pasar la información a la siguiente página y poder mostrarla cuando terminemos la compra.

```
private void pickerProductos_SelectedIndexChanged(object sender, EventArgs e)
{
    var picker = (Picker)sender;
    int selectedIndex = picker.SelectedIndex;
    if (selectedIndex != -1)
    {
        cadena = (string)picker.ItemsSource[selectedIndex];
    }
}
```

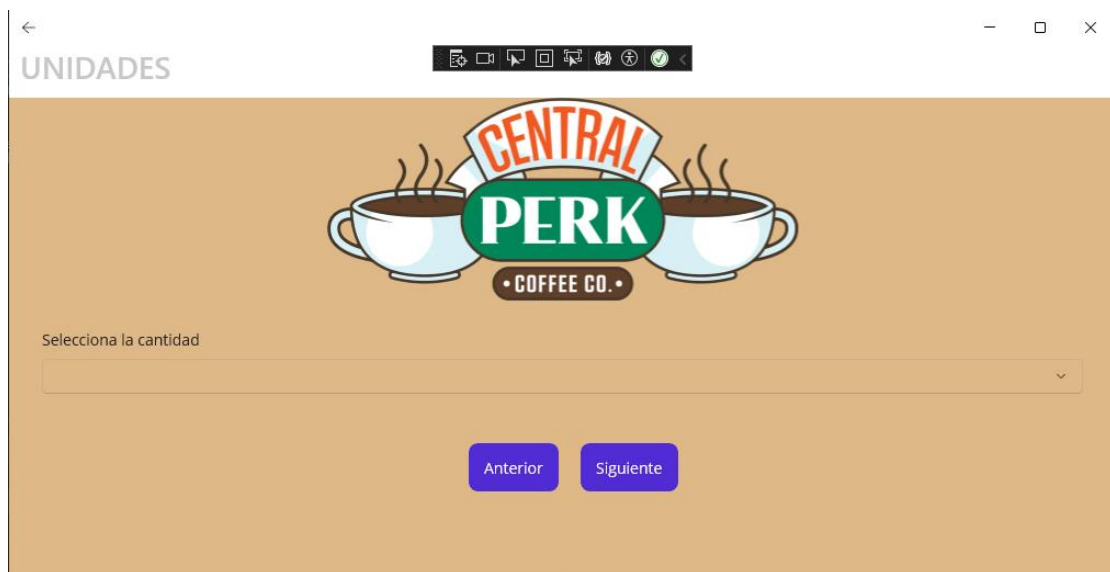
```

public partial class Page2 : ContentPage
{
    String cadena = "";
    public Page2()
    {
        InitializeComponent();
        this.cadena = cadena;
    }

    private async void btSiguienteP2_Clicked(object sender, EventArgs e)
    {
        await Navigation.PushAsync(new Page3(cadena));
    }
}

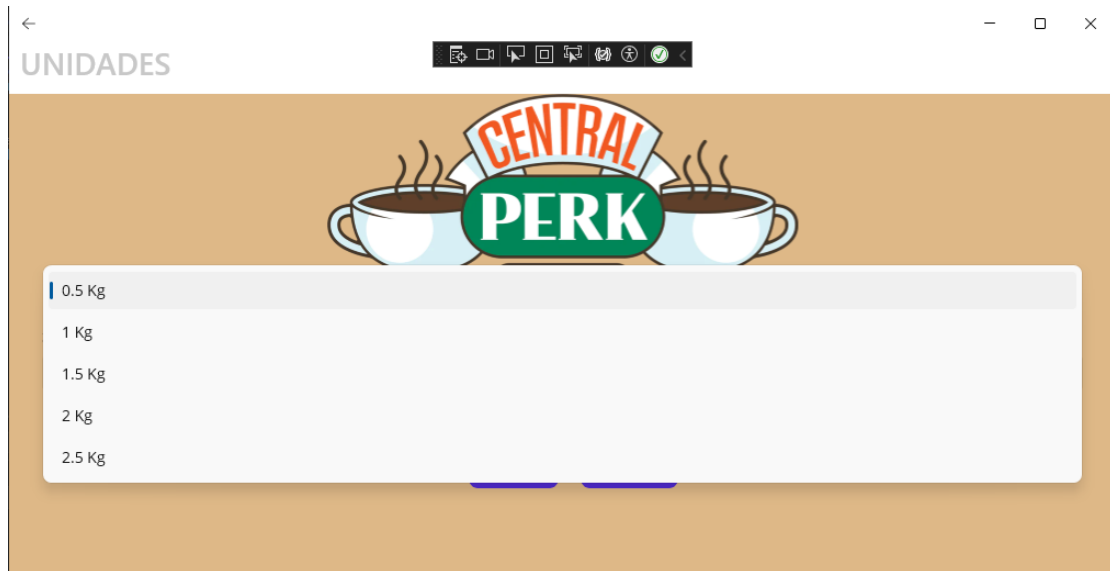
```

## Pag3.xaml



The screenshot shows a mobile application interface with a light brown background. At the top, there is a navigation bar with a back arrow on the left and window controls (minimize, maximize, close) on the right. Below the navigation bar, the word "UNIDADES" is displayed in a light gray font. In the center of the screen is a logo for "CENTRAL PERK COFFEE CO." featuring two white coffee cups with steam rising from them, flanking a green oval with the word "PERK" in white, and a banner above it with the word "CENTRAL" in red. Below the logo, the text "Selecciona la cantidad" is followed by a dropdown menu. At the bottom of the screen, there are two blue buttons: "Anterior" and "Siguiente".






## Page4.xaml

En esta página el usuario tiene que ingresar su dirección

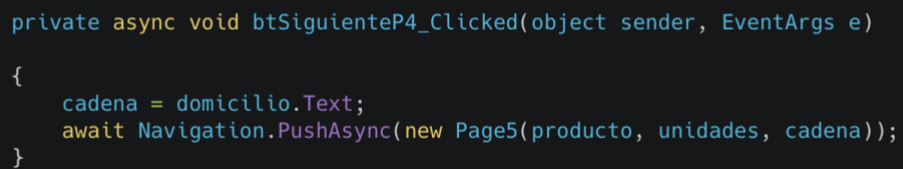




```
<Entry x:Name="domicilio"></Entry>
```

### Page4.xaml.cs

Observamos que, al pulsar el botón Siguiente en la página 4, se guarda la cadena de texto introducida por el usuario.



```
private async void btSiguienteP4_Clicked(object sender, EventArgs e)
{
    cadena = domicilio.Text;
    await Navigation.PushAsync(new Page5(producto, unidades, cadena));
}
```

## Page5.xaml.cs

A continuación, vemos todas las cadenas de texto que el usuario ha seleccionado previamente y las concatenamos, y las guardamos en un label, como podemos apreciar en la imagen.



```
public Page5(String producto, String unidades, String domicilio)
{
    InitializeComponent();
    lbPedido.Text = "Tipo de Cafe: " + producto + "\nUnidades: " + unidades + "\nDirección: " + domicilio;
}
```

## Page5.xaml

```
<Label x:Name="lbPedido"></Label>
```

## Repositorio GitHub

<https://github.com/CevicheConAji/CompraNavigationPage.git>

## Webgrafía

<https://learn.microsoft.com/es-es/dotnet/maui/?view=net-maui-8.0>